

Recursieve algoritmen
Dynamische Programmeren

bottom-up
top-down

Dynamische Programmeren

Afhankelijkheden van deelproblemen

- Verdeel-en-heers: dat functioneert, als de deelproblemen onafhankelijk zijn.
- Als de deelproblemen niet onafhankelijk zijn, kunnen recursieve programma's veel meer tijd nodig hebben.
- Als de deelproblemen NIET onafhankelijk zijn, kan men beter andere systematische technieken toepassen.

Fontys

STL,ODE/FHTBM Recursion 13 november 2007 17/22

Recursieve algoritmen
Dynamische Programmeren

bottom-up
top-down

Fibonacci nog een keer

Het volgende recursieve Fibonacci programma is bijvoorbeeld zeer inefficiënt (en moet niet worden gebruikt):

```
static int F(int i)
{
    if (i<1) return 0;
    if (i==1) return 1;
    return F(i-1)+F(i-2);
}
```

Fontys

STL,ODE/FHTBM Recursion 13 november 2007 18/22

Recursieve algoritmen
Dynamische Programmeren

bottom-up
top-down

Een lineair programma

- Een lineair programma dat een array vult onder gebruikmaaking van een recurrente betrekking is echter snel en gemakkelijk.
- Het bijhouden van de laatste twee fibonacci getallen is voldoende om de berekening uit te kunnen voeren. (exercise 5.37)

Fontys

STL,ODE/FHTBM Recursion 13 november 2007 19/22

Recursieve algoritmen
Dynamische Programmeren

bottom-up
top-down

Dynamisch programmeren bottom-up

- Dynamisch programmeren bottom-up
 - Dit geldt voor elke recursieve berekening, onder de voorwaarde dat we eerder berekende waarden kunnen opstaan. Bijvoorbeeld voorhet fibonacci voorbeeld:

```
F[0] = 0; F[1] = 1;
for (int i=2; i<N; i++)
    F[i]=F[i-1]+F[i-2];
```

Fontys

STL,ODE/FHTBM Recursion 13 november 2007 20/22
